



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi

Metode yang digunakan dalam Implementasi Metode *Local Color Histogram* dan GLCM pada aplikasi temu kenali citra satwa langka berbasis konten, yaitu sebagai berikut.

1) Studi Literatur

Dalam studi literatur, dilakukan pembelajaran mengenai teori atau literatur yang berhubungan dengan *Content Based Image Retrieval (CBIR)*, *Color Histogram*, *Color Quantization*, *Gray Level Co-Occurrence Matrix (GLCM)*, *Euclidean Distance*, *Recall and Precision* dan Satwa Langka. Pembelajaran dilakukan dengan membaca dari berbagai sumber seperti buku, makalah atau jurnal, skripsi, serta berbagai sumber lainnya baik online maupun media cetak.

2) Pengumpulan Data

Dalam penelitian ini, dilakukan pengumpulan *dataset* satwa langka berupa gambar 12 satwa yang ada pada batasan masalah dengan ukuran 384 x 256 piksel dan berformat .JPG serta *dataset* Wang yang telah di-download pada situs <http://wang.ist.psu.edu/docs/related/>. Kedua *dataset* tersebut digunakan pada saat uji coba untuk menghitung nilai *recall* dan *precision*.

3) Perancangan Sistem

Perancangan yang dilakukan pada tahap ini adalah merancang alur kerja sistem yang digambarkan menggunakan diagram alur proses (*flowchart*).

Setelah itu dilanjutkan dengan merancang tampilan atau antarmuka aplikasi serta fitur-fitur yang akan disediakan pada aplikasi ini.

4) Implementasi Sistem

Implementasi sistem adalah realisasi atau pembuatan sistem yang telah dirancang sebelumnya. Implementasi sistem berbasis web dan akan dibangun dengan menggunakan bahasa pemrograman PHP dan *database* MySQL.

5) Uji Coba dan Evaluasi

Sistem yang telah dibuat, diuji pada tahap ini untuk memastikan semua fungsionalitas yang telah ditentukan dapat berjalan dengan baik. Uji Coba dilakukan pada *dataset* satwa langka dan *dataset* Wang. *Dataset* satwa langka memiliki 180 gambar yang terdiri dari 12 kategori dengan masing-masing 15 gambar tiap kategori dan *dataset* Wang memiliki 250 gambar yang terdiri dari 10 kategori dengan masing-masing 25 gambar. Evaluasi dilakukan dengan cara menghitung nilai *recall* dan *precision* pada *dataset* satwa langka dan *dataset* Wang.

6) Penulisan Laporan

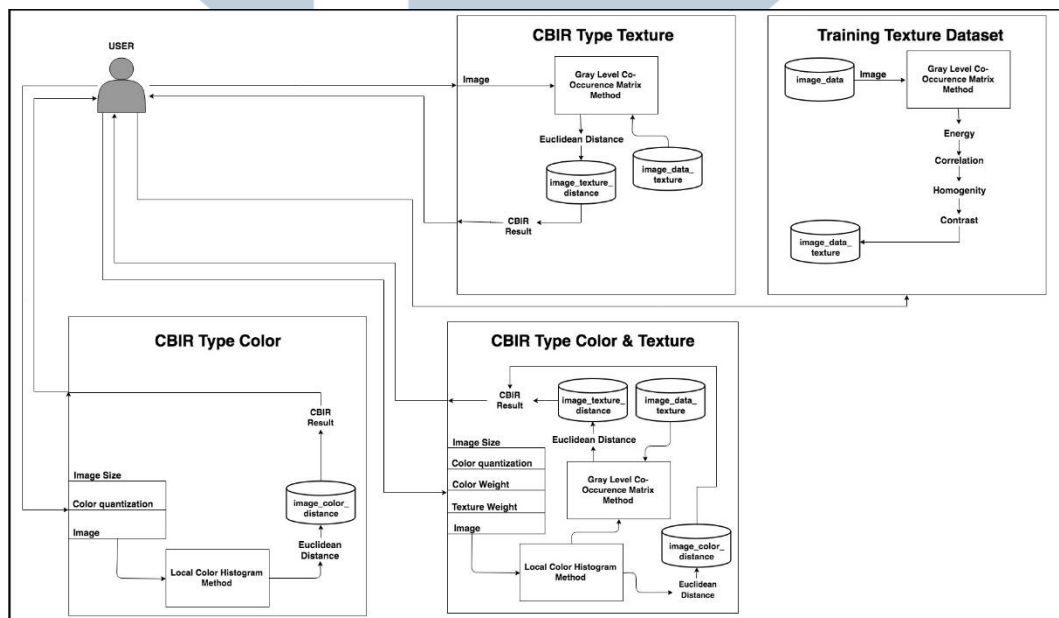
Penulisan laporan bertujuan sebagai dokumentasi dari sebuah penelitian yang telah dilakukan sehingga dapat memberikan informasi bagi penelitian sejenis.

3.2 Perancangan Program

Program dirancang dengan menggunakan *application model*, *flowchart*, dan desain tampilan tampilan antarmuka.

3.2.1 Application Model

Model aplikasi digambarkan pada Gambar 3.1. Pada proses CBIR *type color*, *user* akan memasukkan *image size*, *color quantization*, dan gambar yang diinginkan oleh *user* untuk proses *image retrieval*. Sebelum *user* menjalankan proses CBIR *type texture*, *user* akan menjalankan proses *training texture dataset* untuk mendapatkan fitur tekstur seluruh gambar yang ada dalam *database*. Pada proses CBIR *type texture*, *user* hanya akan memasukkan gambar saja. Namun pada proses CBIR *type color & texture*, *user* akan memasukkan *image size*, *color quantization*, *color weight*, *texture weight*, dan gambar. Pada ketiga proses CBIR tersebut akan mengembalikan daftar 10 gambar dengan urutan kemiripan dari yang paling mirip dengan gambar yang dimasukkan oleh *user*.



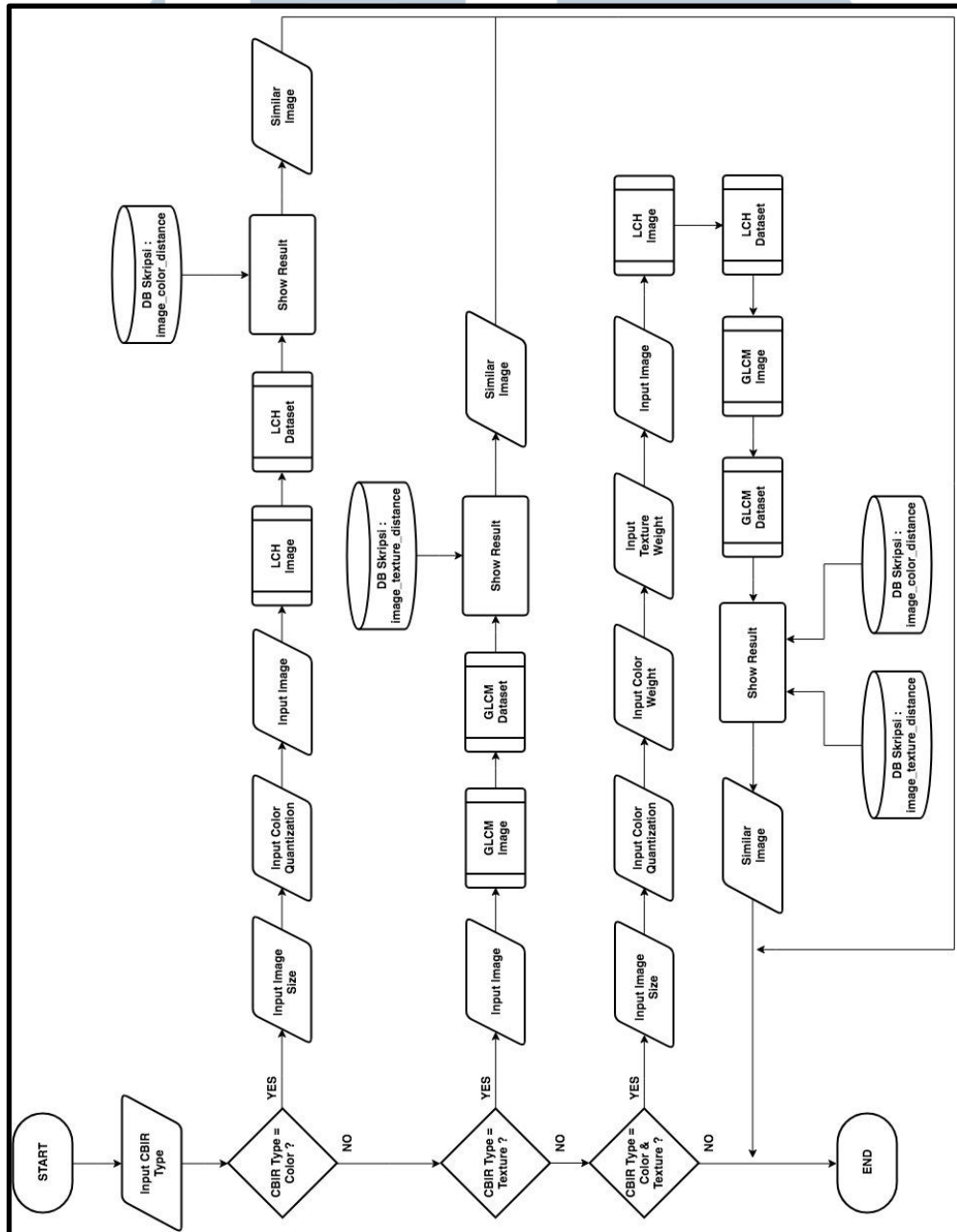
Gambar 3.1 Application Model

3.2.2 Flowchart

Berikut merupakan *flowchart* dari aplikasi temu kenali citra satwa langka berbasis konten.

A. Flowchart Utama

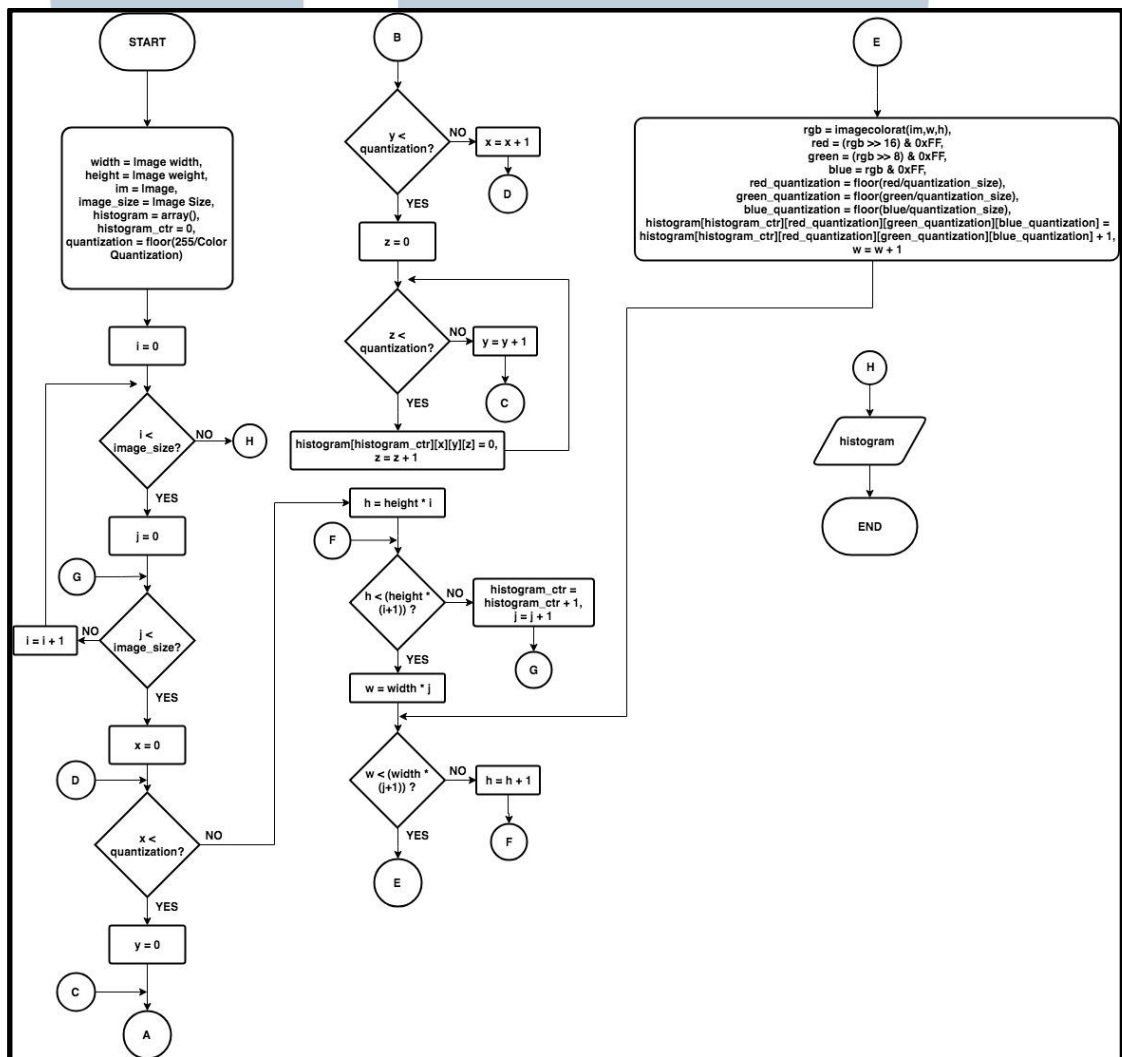
Flowchart Utama digambarkan pada Gambar 3.2. *User* akan diminta untuk memilih tipe CBIR, jika *user* memilih *color* maka *user* akan diminta untuk memasukkan *image size*, *color quantization*, dan *upload* gambar yang ingin dicari di dalam *dataset* dengan ukuran 384 x 256 piksel dan berformat .JPG/.JPEG. *Image size* adalah berapa banyak gambar akan dibagi menjadi bagian yang sama. *Color quantization* adalah kuantisasi warna untuk warna *red*, *green*, dan *blue* agar proses perhitungan histogram menjadi lebih cepat. Apabila *user* memilih tipe CBIR *texture*, maka *user* akan diminta untuk melakukan *upload* gambar yang ingin dicari di dalam *dataset*. Lalu, jika *user* memilih tipe CBIR *color & texture*, maka *user* diminta untuk memasukkan *image size*, *color quantization*, *color weight*, *texture weight*, dan *upload* gambar yang ingin dicari di dalam *dataset*. *Color weight* merupakan bobot untuk fitur warna, sedangkan *texture weight* adalah bobot untuk fitur tekstur. Total dari kedua bobot adalah 100 persen, bobot berfungsi sebagai perbandingan besar penerapan metode *local color histogram* dan metode *gray level co-occurrence matrix*. Sebagai contoh bobot untuk *color* sebesar 30 persen dan *texture* sebesar 70 persen, maka hasil *euclidean distance* akhir adalah 30 persen dikali dengan *euclidean distance* yang diperoleh dari metode *local color histogram* ditambah dengan 70 persen dikali dengan *euclidean* yang diperoleh dari metode *gray level co-occurrence matrix*.



Gambar 3.2 Flowchart Utama

B. Flowchart LCH Image

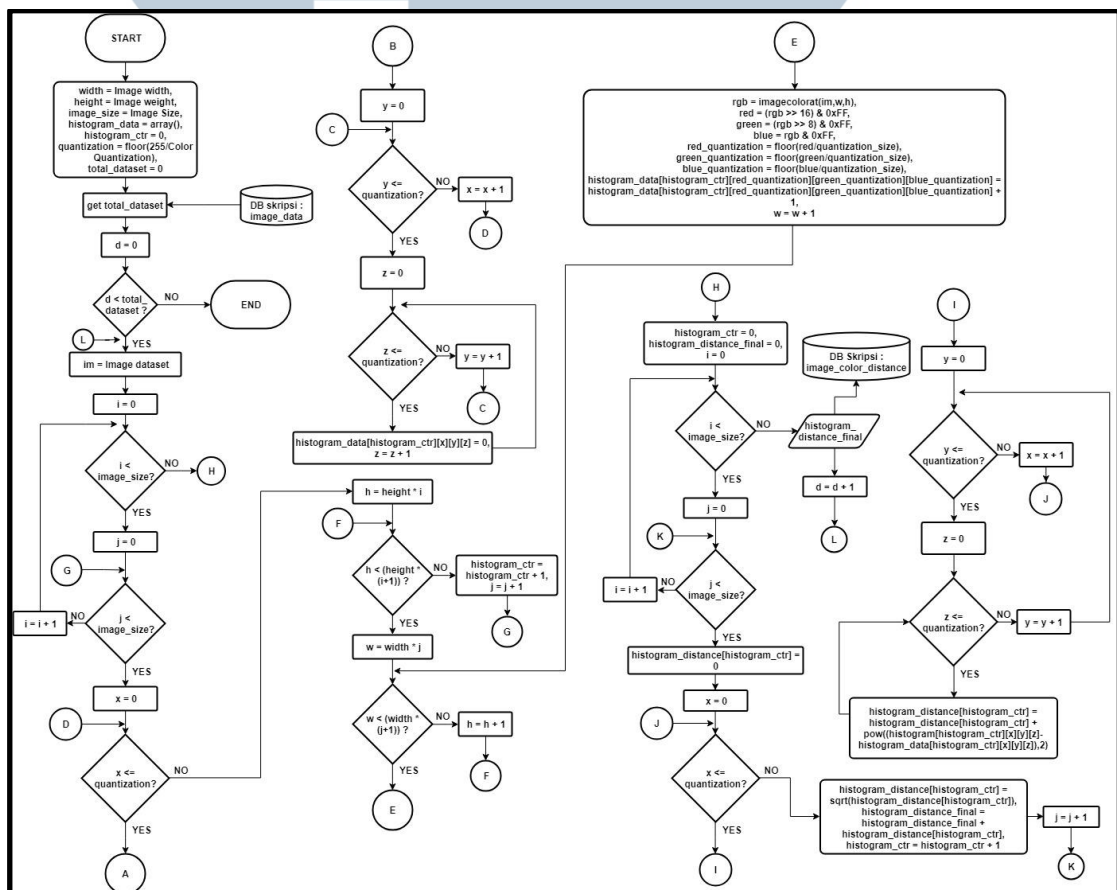
Flowchart LCH image menjabarkan proses pencarian *histogram* dari gambar yang telah di *upload* oleh user. Gambar tersebut akan dibagi menjadi beberapa bagian dengan ukuran yang sama sesuai dengan *image size* yang telah dimasukkan oleh *user*. Kemudian, masing-masing dari bagian tersebut akan dicari *histogram*nya dengan kuantisasi warna sebesar *color quantization* yang sudah dimasukkan oleh user sebelumnya. Gambar 3.3 menjabarkan *flowchart LCH image*.



Gambar 3.3 *Flowchart LCH Image*

C. Flowchart LCH Dataset

Flowchart LCH dataset menjabarkan proses pencarian *histogram* dari gambar yang ada pada *dataset*. Langkah pertama yaitu sistem akan mengambil semua data yang ada pada *dataset* di dalam tabel *image_data*. Kemudian, untuk setiap data yang ada pada *dataset* akan dicari *histogram*nya sesuai dengan *image size* dan *color quantization* yang sudah dimasukkan oleh *user* sebelumnya. Setelah *histogram* setiap data pada *dataset* diperoleh, maka akan diukur jarak dari *histogram* setiap data tersebut dengan *histogram* dari gambar yang telah di-upload oleh *user* menggunakan *euclidean distance*. Gambar 3.4 menjabarkan *flowchart LCH dataset*.



Gambar 3.4 Flowchart LCH Dataset

D. Flowchart GLCM Image

Flowchart GLCM Image menjabarkan proses pengekstrakan fitur tekstur dari gambar yang telah di-upload oleh user. Warna gambar akan diubah ke *grayscale* terlebih dahulu, setelah itu dilanjutkan dengan pencarian matriks GLCM untuk sudut 0, 45, 90, dan 135 serta *transpose* dari matriks GLCM sudut 0, 45, 90, dan 135. Kemudian matriks GLCM dan matriks *transpose* yang telah didapat saling dijumlahkan untuk setiap sudutnya. Matriks GLCM sudut 0, 45, 90, dan 135 akan di normalisasi dengan cara membagi matriks dengan total penjumlahan matriks GLCM dengan matriks *transpose*. Lalu, matriks GLCM sudut 0, 45, 90, dan 135 akan digabungkan dengan mencari rata-rata dari keempat sudut matriks GLCM tersebut. Setelah gabungan matriks GLCM didapat, maka dilanjutkan dengan mencari *mean* dan *variance* yang akan digunakan untuk menghitung empat fitur dari *Gray Level Co-Occurrence Matrix* yaitu *energy*, *correlation*, *homogeneity/inverse different moment*, dan *contrast*. Gambar 3.6 menjabarkan *flowchart GLCM image*.



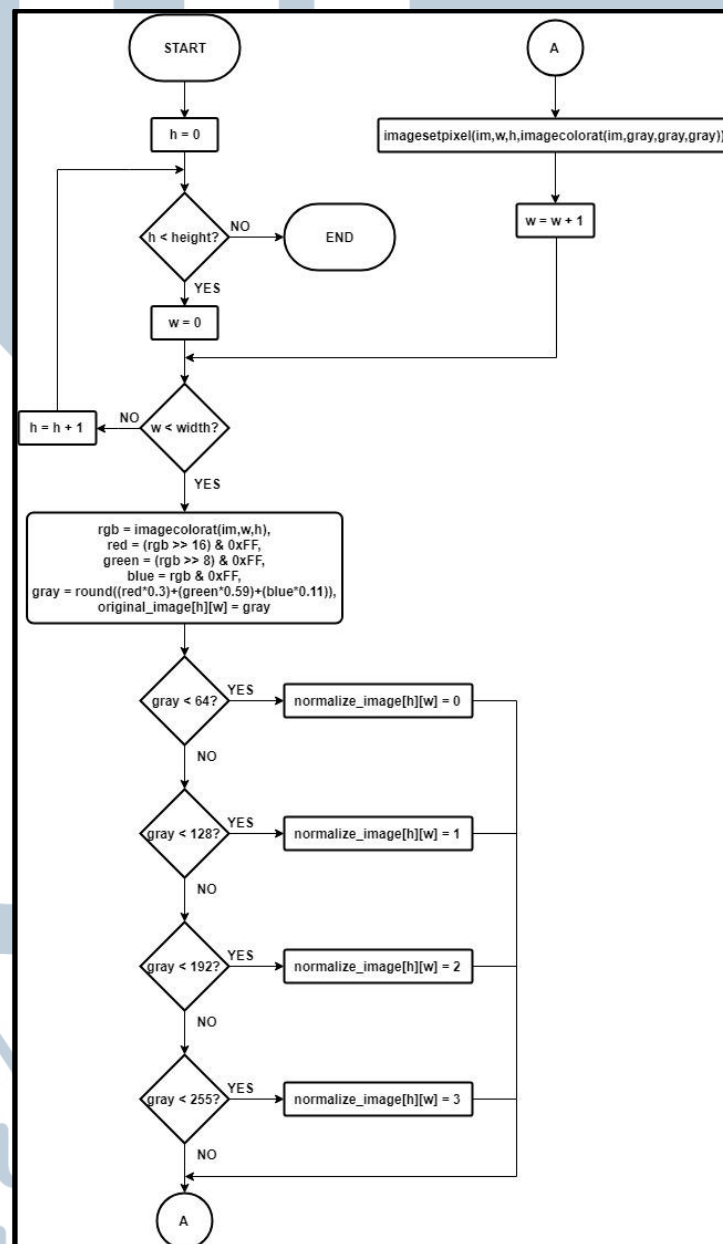
E. Flowchart GLCM Dataset

Flowchart GLCM dataset menjabarkan proses pengekstrakan fitur tekstur dari semua gambar yang ada pada *dataset*. Langkah pertama yaitu sistem akan mengambil semua data yang ada pada *dataset* di dalam tabel *image_data*. Kemudian akan dilakukan pencarian fitur *energy*, *correlation*, *homogeneity/inverse different moment*, dan *contrast* menggunakan metode *Gray Level Co-Occurrence Matrix* untuk setiap data pada *dataset*. Setelah keempat fitur setiap data pada *dataset* diperoleh, maka akan dicari jarak dari keempat fitur tersebut dengan empat fitur yang didapat dari gambar yang telah di *upload* oleh *user* dengan menggunakan *euclidean distance*. Gambar 3.7 menjabarkan *flowchart GLCM dataset*.



F. Flowchart Grayscale Image

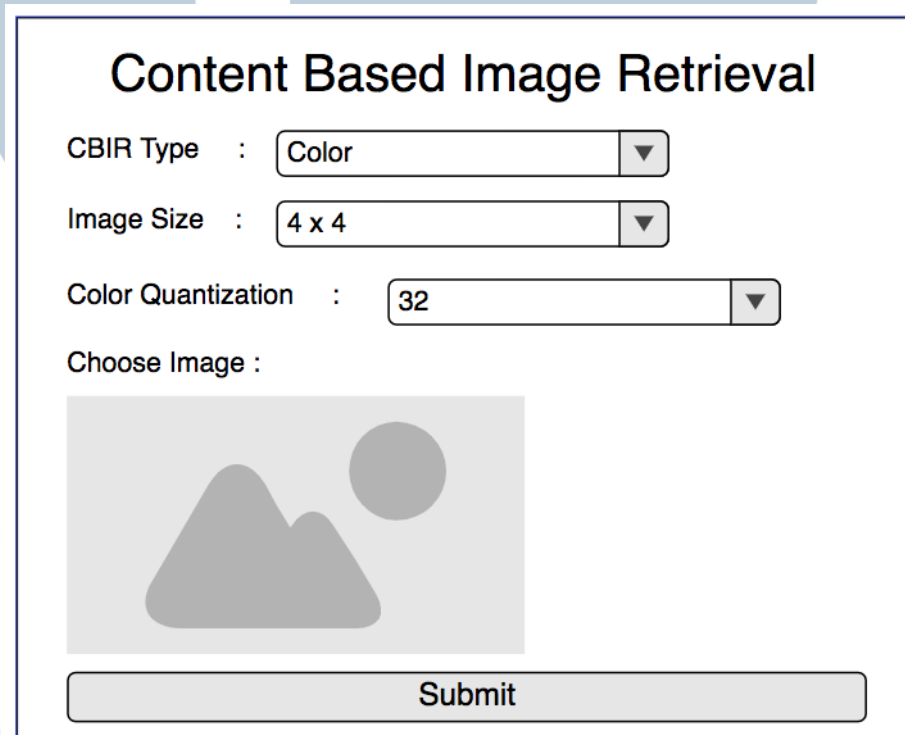
Flowchart grayscale menjabarkan proses konversi warna gambar yang telah di *upload* oleh *user* menjadi warna abu-abu atau *grayscale*. Setelah itu akan dicari normalisasi matriks dengan 4 level keabuan dari gambar tersebut. Gambar 3.5 menjabarkan *flowchart grayscale image*.



Gambar 3.7 Flowchart Grayscale Image

3.2.3 Desain Tampilan Antarmuka

Dalam mengimplementasikan aplikasi temu kenali citra satwa langka berbasis konten dilakukan dengan desain tampilan antarmuka terlebih dahulu. Desain antarmuka pada Gambar 3.8 dirancang untuk halaman pada saat *user* mengakses aplikasi. *User* dapat memilih tipe CBIR yang ingin digunakan, tetapi secara *default* tipe CBIR adalah *color*. *User* dapat memilih *image size* dan *color quantization* yang diinginkan serta melakukan *upload* gambar.




Content Based Image Retrieval

CBIR Type :

Image Size :

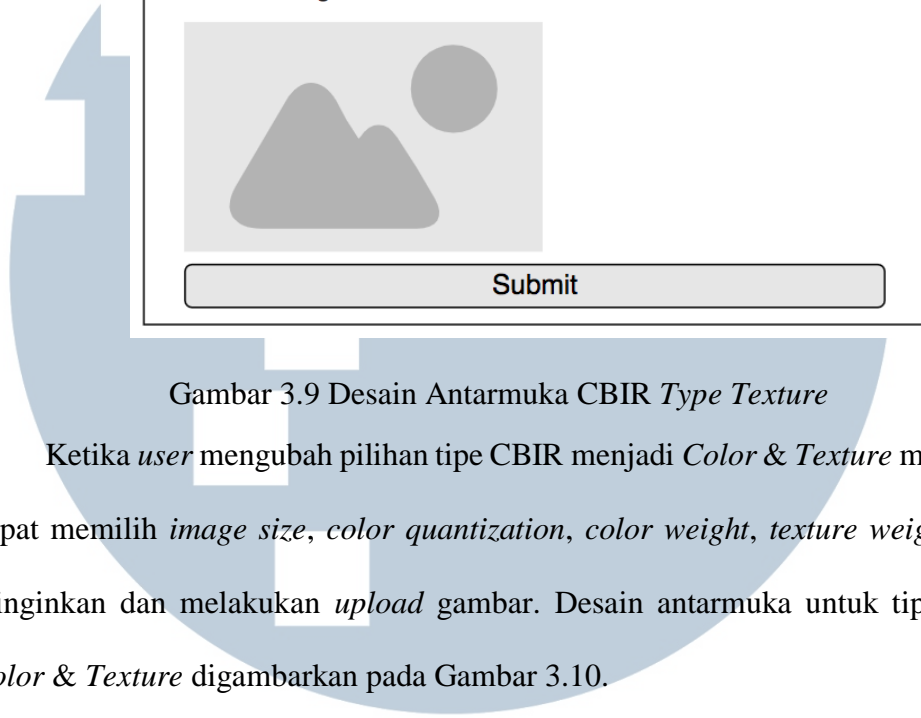
Color Quantization :

Choose Image :



Gambar 3.8 Desain Antarmuka Halaman CBIR Type *Color*


Apabila *user* mengubah pilihan tipe CBIR menjadi *Texture* maka *user* hanya perlu untuk melakukan *upload* gambar saja. Desain antarmuka untuk tipe CBIR *Texture* digambarkan pada Gambar 3.9.



Content Based Image Retrieval

CBIR Type :

Choose Image :



Gambar 3.9 Desain Antarmuka CBIR Type *Texture*

Ketika *user* mengubah pilihan tipe CBIR menjadi *Color & Texture* maka *user* dapat memilih *image size*, *color quantization*, *color weight*, *texture weight* yang diinginkan dan melakukan *upload* gambar. Desain antarmuka untuk tipe CBIR *Color & Texture* digambarkan pada Gambar 3.10.

Content Based Image Retrieval

CBIR Type :


Image Size :

Color Quantization :

Color Weight :

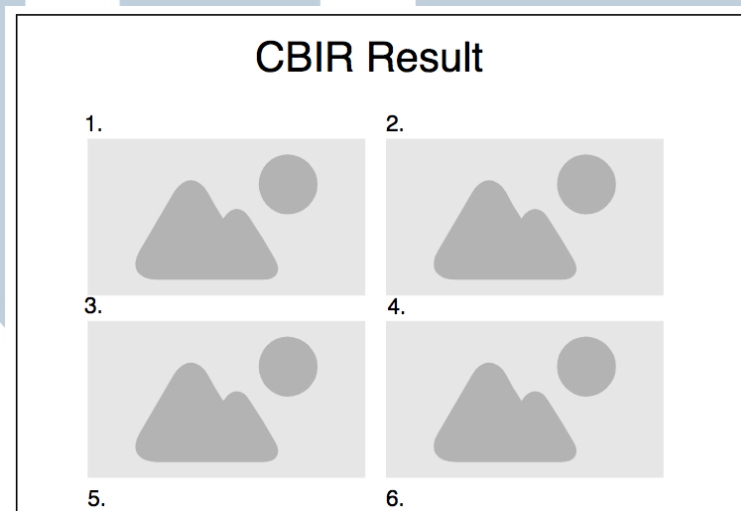
Texture Weight :

Choose Image :

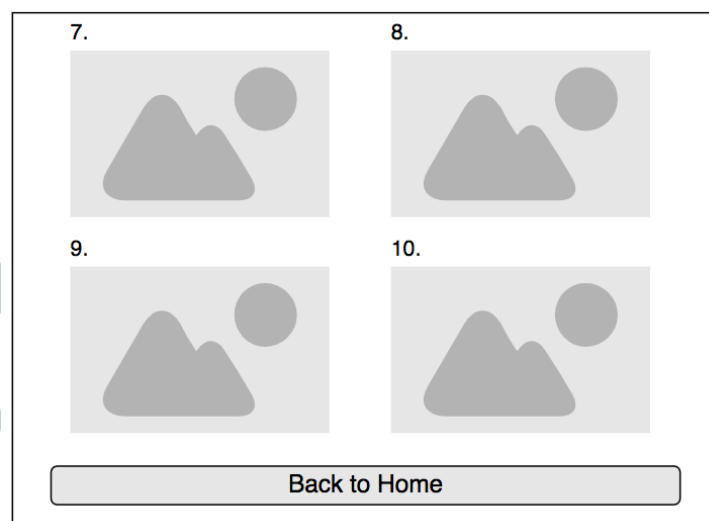


Gambar 3.10 Desain Antarmuka CBIR Type *Color & Texture*

Setelah *user* menekan tombol *submit* maka *user* akan diarahkan ke halaman *CBIR Result* pada Gambar 3.11. Pada halaman *CBIR Result* berisi 10 daftar gambar yang memiliki urutan kemiripan dari yang paling mirip dengan gambar yang di *upload* oleh *user*. Pada bagian bawah halaman *CBIR Result* ada tombol *back to home* yang digunakan *user* kembali ke halaman awal pada saat *user* mengakses aplikasi yang digambarkan pada Gambar 3.12



Gambar 3.11 Desain Antarmuka *CBIR Result*



Gambar 3.12 Desain Antarmuka *CBIR Result*